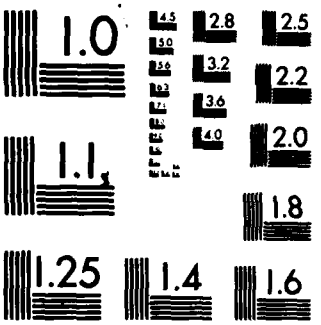
AD-A128 661 ADAPTING A CONSULTATION SYSTEM TO CRITIQUE USER PLANS 1/1
(U) STANFORD UNIV CA HEURISTIC PROGRAMMING PROJECT
C P LANGLOTZ ET AL. APR 83 HPP-83-2 N00014-81-K-0004
UNCLASSIFIED F/G 9/2 NL

END
DATE FILMED
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Stanford Heuristic Programming Project
Memo HPP-83-2

April 1983

Adapting a Consultation System to Critique User Plans

by

Curtis P. Langlotz

and

Edward H. Shortliffe

Departments of Medicine and Computer Science
Stanford University
Stanford, California 94305

Adapting a Consultation System to Critique User Plans

Curtis P. Langlotz
and
Edward H. Shortliffe

Running Title: An Expert System That Critiques Plans

Heuristic Programming Project
Departments of Medicine and Computer Science
Stanford University
Stanford, California 94305
April 1983

To appear in *International Journal of Man-Machine Studies*
(Special Issue on Expert Systems)

Support for this work was provided by the National Library of Medicine under Grants LM-03395 and LM-00048, and by the Office of Naval Research under contract NR 049-479. Dr. Shortliffe is a Henry J. Kaiser Family Foundation Faculty Scholar in General Internal Medicine. Computing facilities were provided by the SUMEX-AIM resource under NIH Grant RR-00785.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A128661	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Adapting a Consultation System to Critique User Plans	5. TYPE OF REPORT & PERIOD COVERED Technical
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Curtis P. Langlotz and Edward H. Shortliffe	8. CONTRACT OR GRANT NUMBER(s) ONR N00014-81-K-0004
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Computer Science Stanford University Stanford, CA 94305 USA	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Personnel and Training Research Programs Office of Naval Research (Code 458) Arlington, VA 22217	12. REPORT DATE April 1983 / 13. NO. OF PAGES 21
	15. SECURITY CLASS. (of this report) Unclassified
14. MONITORING AGENCY NAME & ADDRESS (if diff. from Controlling Office) ONR Representative-Mr. Robin Simpson Durand Aeronautics Building, Room 165 Stanford University Stanford, CA 94305	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A predominant model for expert consultation systems is one in which a computer program simulates the decision making processes of an expert. The expert system typically collects data from the user and renders a solution. Experience with regular physician use of ONCOCIN, an expert system that assists with the treatment of cancer patients, has revealed that system users can be annoyed by this approach. In an attempt to overcome this barrier to system acceptance, ONCOCIN has been adapted to accept, analyze, and critique a physician's own therapy plan. A critique is an explanation of the significant differences between the plan that would have been proposed by the expert system and the plan proposed by the user. The critique helps

DD FORM 1 JAN 73 1473
EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19. KEY WORDS (Continued)

20 ABSTRACT (Continued)

resolve these differences and provides a less intrusive method of computer-assisted consultation because the user need not be interrupted in the majority of cases - those in which no significant differences occur. Extensions of previous rule-based explanation techniques has been required to generate critiques of this type.

Abstract

A predominant model for expert consultation systems is one in which a computer program simulates the decision making processes of an expert. The expert system typically collects data from the user and renders a solution. Experience with regular physician use of ONCOCIN, an expert system that assists with the treatment of cancer patients, has revealed that system users can be annoyed by this approach. In an attempt to overcome this barrier to system acceptance, ONCOCIN has been adapted to accept, analyze, and critique a physician's own therapy plan. A *critique* is an explanation of the significant differences between the plan that would have been proposed by the expert system and the plan proposed by the user. The critique helps resolve these differences and provides a less intrusive method of computer-assisted consultation because the user need not be interrupted in the majority of cases -- those in which no significant differences occur. Extension of previous rule-based explanation techniques has been required to generate critiques of this type.

Introduction

As symbolic reasoning techniques that had developed in the field of artificial intelligence (AI) matured, their potential power as problem solving tools became clear and a number of expert advice systems were developed using AI methods (Duda, 1983). One of the important lessons learned from the early expert consultants is that *excellent decision-making performance does not guarantee user acceptance* (Shortliffe, 1982). This lesson has been particularly evident in the field of medicine where experience applying AI techniques has helped reveal what capabilities can make computer-based consultation systems attractive to users, and what characteristics detract from their acceptability. A consultation program's ability to explain its reasoning has been shown to be particularly important in obtaining user acceptance (Teach, 1981).

In this paper we describe recent additional lessons learned through our work on ONCOCIN, an expert system used by physicians in their routine care of cancer patients. Because this system is used for data management as well as for consultations, physicians do not always require or desire advice when they use the program. This is an important distinction from conventional expert systems in which the program can typically assume that the user is specifically seeking assistance with a decision task. We believe it will be preferable to allow ONCOCIN's users to indicate their own management plans, and to monitor those plans for apparent errors, rather than to generate advice routinely. Early experience suggests that this *critiquing model*[1] of expert system interaction will enhance a program's acceptability for some applications.

The paper begins with a brief description of the ONCOCIN design strategy and explains how the concept of a critiquing model evolved from our recent experience using the system as a conventional consultation program. The program's architecture is described and we then provide a detailed description and examples of our recent experiments in adapting the system for the critiquing model. We conclude with a discussion of our plans to implement the model in the operational system as we adapt ONCOCIN to run on professional workstations rather than the main-frame machine on which it currently operates.

ONCOCIN and the Critiquing Model

ONCOCIN (Shortliffe, 1981) is a medical consultation program which uses knowledge of cancer treatment protocols[2], encoded in production rules, to assist physicians with therapy decisions for cancer patients. It is designed to provide excellent decision-making performance while addressing a number of acceptability issues. Thus, although our main purpose is to provide high-quality management advice, we have tried to develop a system that physicians can use directly, and that they consider both helpful and suitable for regular use.

ONCOCIN is also designed to avoid additive time demands on the already busy schedule of an oncologist (cancer specialist). Rather than requiring the physician to perform a new task, it *replaces* one that many oncologists already performed manually, namely that of filling out patient data forms. Before the introduction of ONCOCIN, clinic oncologists were routinely required to fill in a time-oriented record, called a "flowsheet," which is used to maintain patient data for analysis of the effectiveness of alternate therapy regimens. Each row in the paper flowsheet corresponds to a particular test or finding. For each patient-visit,

---

[1] To our knowledge, the term *critiquing model* was first used by Perry Miller (Miller, 1983) in his work on the ATTENDING system described later in this paper.

[2] Protocols are detailed documents that specify alternate therapies to be compared in a formal experiment. In particular, they specify guidelines for delaying or modifying treatments according to patient response. They also define the data that need to be collected in order to provide an adequate basis for judging the merits of the alternate therapy plans.

the physician enters relevant signs, symptoms, and laboratory data in the column on the flowsheet corresponding to that visit. ONCOCIN allows these entries to be made at a computer terminal instead. Consequently, ONCOCIN's computer records function as the primary source of protocol data for that patient. This requires that the physicians use ONCOCIN for *all* visits of patients whose records are on the computer, but avoids the need (and frequent inaccuracy) of clerical transcription of flowsheet data into statistical analysis routines. The availability of on-line data also allows ONCOCIN to print out automatically a variety of paper reports which physicians were accustomed to preparing by hand. These features serve to reduce the amount of time physicians must spend performing routine tasks and have heightened ONCOCIN's acceptance by its users.

We introduced the prototype system in our outpatient oncology clinic in May 1981. Physicians were involved from the outset in the system's design, and we have received valuable additional feedback from them since its introduction. The most frequent complaint raised by physicians who use ONCOCIN is that they become annoyed with changing or "overriding" ONCOCIN's treatment suggestions. Physicians override ONCOCIN's decision if they disagree, even slightly, with the program's treatment recommendation and choose to give an alternate treatment instead. Each time this occurs, the physician is asked to provide a justification for any changes. The "override" feature allows the physician to remain the final decision-maker regarding the treatment given to a patient. However, if the user must frequently override ONCOCIN, this can be irritating and time-consuming, particularly when the changes are minor and reflect slight dosing adjustments for patient convenience or to increase compliance with the regimen.

It occurred to us that if the consultation system were modified to monitor and critique the therapy plans proposed by users, it could conduct a consultation in a much less disruptive manner. In addition, it would allow the doctor routinely to suggest treatment first and thereby to remain more actively involved in the consultation process. The educational role of the system would also be heightened because a critiquing system could not only point out the differences between its own recommendation and the physician's, but it could also help the physician make an informed choice between the two by explaining the differences.

Little would be gained by such an approach if the consultation program were to enter into a lengthy critique of *every* plan entered by the user. However, if there were only minor differences between the physician's proposed therapy and the optimal therapy determined by the consultation system, the physician would not need to be bothered by the expert decision-making portion of the program. Only when a significant disagreement occurred would ONCOCIN interrupt to explain the problem it had noted.

An additional appeal of the critiquing approach was that it would facilitate clearer explanations when they were needed. The user's analyzed therapy plan could be used to focus the discourse because it would provide an accurate indication of the areas of knowledge of importance to the user. The goals of the explanation dialogue would thereby be more clear than if the user had simply asked a general question of the system. Measures of what is important to a user are difficult to determine in other ways (Wallis, 1982).

The requirement that a system explain differences between its own recommendation and that of the user places the development of the critiquing model within the area of explanation research. Because disagreements occur often, even among experts (Yu, 1979), it is not surprising that many system users consider an ability to explain the advice the single most important feature for computer-based clinical consultation systems. This is certainly the case for diagnostic or treatment advice systems designed for physicians (Teach, 1981). When disagreements between the computer and user occur, explanation can provide assurance that the computer's reasoning is logical and that unexpected advice is appropriate (Scott, 1977). Some existing expert systems (Clancey, 1981; Shortliffe, 1976; Swartout, 1981) have tried to address

this demand by providing explanations or tutorial facilities during and after a consultation. However, the therapy critiquing system developed for MYCIN (Clancey, 1977) and Miller's ATTENDING program (Miller, 1983) are, to our knowledge, the only previous programs designed to resolve conflicts between a computer-based consultant's advice and a plan preferred by the program's user. Clancey's critiquing system offered only a composite critique for each proposed therapy recommendation. It was unable to give an analysis in varied levels of detail, or to select appropriate sub-parts of MYCIN's reasoning process to explain to the user. ATTENDING critiques a preoperative anesthetic plan using knowledge of anesthesia in a decision network of anesthetic procedures and their associated risks. Risks associated with alternate procedures in the network are used to generate a prose analysis which compares the relative risks and benefits of the user's proposed plan with those of alternate strategies. As with Clancey's system, ATTENDING offers a summary analysis; users are unable to ask for analyses of particular parts of their proposed plans.

By enabling ONCOCIN to analyze and critique physicians' therapy plans, we have tried to prevent the irritations of frequent physician overrides. The system allows the doctor to routinely suggest treatment first[3], then interrupts only if it detects a significant difference between the doctor's and its own recommendation. It is able to relax the constraints under which it identifies conflicts with the doctor's plan; consequently, the user can prescribe without being interrupted a therapy plan that is clinically acceptable within a tolerance range of the computer's plan. The physician retains the initiative, but ONCOCIN's advisory capabilities are still available whenever they are needed.

ONCOCIN's System Architecture

ONCOCIN is currently implemented on a time-shared computer which allows us to structure the system as two separate programs running in parallel: the *Interviewer*, a display program which manages the terminal used by the doctors, and the *Reasoner*, a rule-based AI program which makes decisions about the treatment of a patient according to the data it receives from the *Interviewer* (Gerring, 1982).

The Interviewer

ONCOCIN allows the doctor to enter patient data directly into the computer through the *Interviewer*, a specially designed, display-oriented interface that mimics the format of the familiar paper flowsheet. After seeing each patient, the doctor uses the computer terminal, controlled by the *Interviewer*, and communicates via a simple control keypad that moves the cursor to arbitrary locations on the computer flowsheet for data entry. Fig. 1 shows an example of what the terminal screen looks like during a typical session with the physician. The *Interviewer* screen is divided into sections. At the bottom of the screen is a row of "soft key" descriptors. These correspond to keys labeled with roman numerals located across the top of the specialized keyboard. The *Interviewer* assigns different meanings to these keys depending on which portion of the flowsheet is currently displayed. In Fig. 1, for example, when the key labeled "VI" (corresponding to "CHANGE OLD DATA") is pressed, the system enters a mode which allows the physician to change or update data from previous visit dates. At another time in the session, key VI might have a different label and effect.

---

[3] It should be noted that nothing in the critiquing process *requires* that the physician enter a proposed recommendation before seeing the computer's recommendation. In fact, some users might prefer to use the computer's recommendation as a guide, and to enter only their proposed changes.

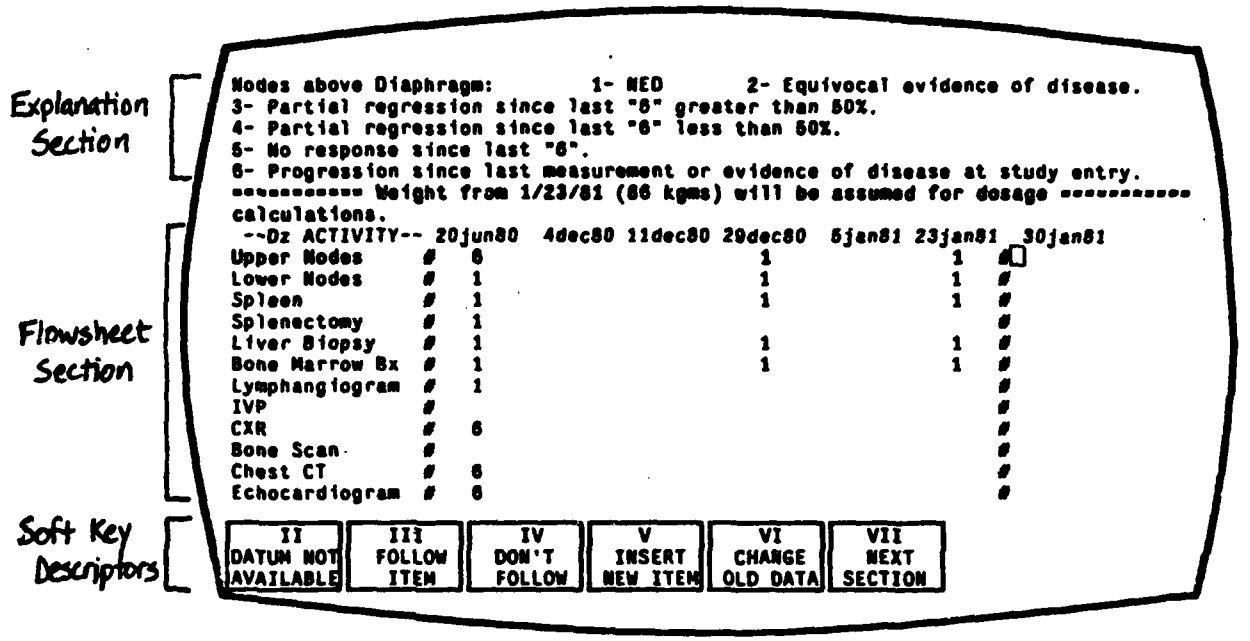


Figure 1: A sample explanation from ONCOCIN. In the explanation section of the screen is an explanation for the "Upper Nodes" disease activity datum on the flowsheet. The square box indicates the cursor location on the screen.

The central portion of the *Interviewer* screen is the "flowsheet" section. In Fig. 1 the disease activity section of the flowsheet is shown (labeled "Dz Activity"). Each column is labeled at the top with a patient visit date, and each row is labeled on the left with the name of the particular finding or test result to which it corresponds. The right-most column corresponds to the current visit date for which the physician is entering data.

Above the computer-based flowsheet is the "explanation" section. Explanations are displayed there automatically according to the location of the cursor on the flowsheet. Two kinds of explanations may be displayed: explanations of what data the system is expecting (during the data entry process) or brief explanations of the program's reasoning (when the cursor is aligned with part of a recommendation). In the data entry display of Fig. 1, for example, the cursor rests in the row corresponding to the "Upper Nodes" item. Since this is an item for which the physician is expected to provide the response, the explanation field provides more detailed information about the item, and makes clear what types of responses are expected. In this case the explanation indicates that the response entered for the "Upper Nodes" disease activity datum may range between *1* and *6*: *1* signifies "No Evidence of Disease" (NED), *2* signifies equivocal evidence of disease, etc.

As the interaction continues, the physician uses the specialized keypad to move the cursor down the flowsheet. When the physician reaches the place on the computer flowsheet where he or she would normally have written the therapy plan on the paper flowsheet, it has already been filled in with the treatment recommendation derived by the *Reasoner* using rules that represent both the protocol knowledge and the judgmental knowledge acquired from our collaborating experts. As shown in Fig. 2, the second kind of explanation is shown for those data items filled in by the *Reasoner*. In this example, since the cursor rests in

the row for the *Reasoner*'s suggested dosage of the drug "Velban", a brief justification of the computer's therapy recommendation is given.

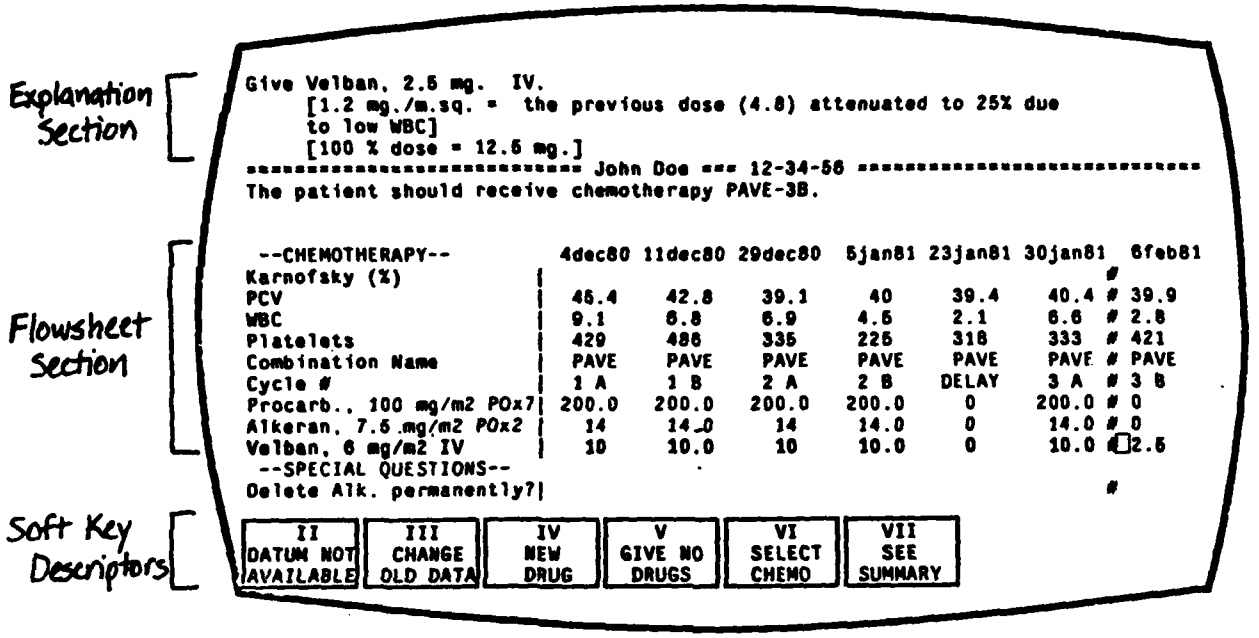


Figure 2: A display on which the therapy recommendations of the *Reasoner* have been displayed. In contrast to Fig. 1, this time the *Explanation Window* shows the system's justification of the therapy advice corresponding to the user-controlled cursor location (box).

The Reasoner

The *Reasoner* uses production rules that encode specific knowledge of chemotherapy protocols as well as general strategies of oncology chemotherapy. These rules, together with data about the patient being treated, are used to make therapy recommendations using both data- and goal-driven reasoning[4]. Each datum, as it is received from the *Interviewer*, is passed to the *Reasoner* which can use it when formulating a therapy recommendation. In a typical consultation, ONCOCIN determines whether the patient should receive chemotherapy on this visit, have therapy delayed, or have the current cycle aborted[5]. If the patient can be treated, then ONCOCIN determines the appropriate dose for each drug in the chemotherapy to which the current patient is assigned. The proper dose is determined by first deciding whether each drug should be omitted. If a drug should not be omitted, it determines the appropriate level of dose attenuation (i.e., dose reduction) or escalation, if any. The result of the reasoning process is a comprehensive chemotherapy treatment plan.

---

[4] See (Shortliffe, 1981) for a more detailed description of ONCOCIN's control structure.

[5] Cancer chemotherapy is often given in "cycles" in which drugs are given on two days a week apart (the "A" and "B" cycle); then a long pause of 3 or 4 weeks is taken before the next cycle begins. A cycle is "aborted" if, after an "A" cycle, no "B" cycle is given because of toxic reactions to therapy.

The Critiquing Process

We describe here our work in developing a prototype critiquing module for ONCOCIN. The capability is not yet part of the system being used in the clinic because we intend to refine it and make use of graphical input and display as ONCOCIN is transferred to run on personal LISP machines. Thus we plan to use the critiquing model as the primary mode of interaction in the new hardware environment to which our physician-users will be introduced over the next few years. Until then ONCOCIN will continue to suggest therapy for all patients as was described above.

In the first step of the prototype critiquing process, the physician enters patient data on the computer flowsheet. Using the data, ONCOCIN formulates a recommendation. Instead of showing the recommendation to the user, however, it is withheld until after he or she has entered a proposed recommendation. The critiquing process then uses an evaluation process which systematically compares the physician's plan to the one formulated by ONCOCIN. Explanations of clinically significant differences are generated if the physician requests them. Once satisfied, the physician can 1) accept ONCOCIN's recommendation, 2) modify ONCOCIN's recommendation, or 3) choose to follow his or her original recommendation. The following sections describe in detail the critiquing techniques used.

Hierarchical Plan Analysis

It has been shown that comparison with an expert can be an effective evaluation method (Burton, 1979). Thus, after the physician's recommendation is obtained, the therapy plan is evaluated by comparing it to the computer's recommendation. The comparison process uses domain specific knowledge of the components of a recommendation and their inter-dependencies. For example, ONCOCIN's recommendation hierarchy shown in Fig. 3 indicates that decisions about radiation therapy are dependent on the choice of chemotherapy, and that decisions about the dose adjustments of a drug are dependent on whether that agent is to be omitted. These types of hierarchical relationships frequently exist in clinical medicine (see (Patil, 1981) for another example). At each level of the hierarchy, ONCOCIN determines whether two analogous components of the computer's and physician's recommendations differ in a clinically significant way using an evaluation procedure specifically designed for that component of a recommendation. For example, in ONCOCIN, we obtained drug-specific dosage tolerances from an expert oncologist; they are used to determine whether a clinically significant difference exists between two different drug doses[6].

Hierarchical plan analysis is an evaluation process designed to find the most general set of differences which completely account for the significant disagreements between two therapy plans. An analysis begins at the most general component in the hierarchy. Analogous components from each recommendation are compared using the corresponding evaluation procedure. Thus, an evaluation begins with the comparison of chemotherapies. If they are in agreement, the sub-components of *Chemotherapy* (*Radiation therapy* and *Circumstance*[7]) are each evaluated. If a clinically significant difference is found in *all* such sub-components, then all differences have been accounted for and explanations will be generated for these sub-components. If

---

[6] A challenging topic for future research is to model the expert's knowledge that allows him to calculate reasonable tolerances for the various drugs used in cancer therapy. This would allow ONCOCIN to determine dynamically the significance of dosing differences and to better explain its basis for deciding that the physician's plan involves an inappropriately high or low dose. Currently it can only quote the tolerance provided by our collaborating expert.

[7] *Circumstance* is the parameter whose value determines whether the patient should receive chemotherapy on this visit, have therapy delayed, or have the current cycle aborted.

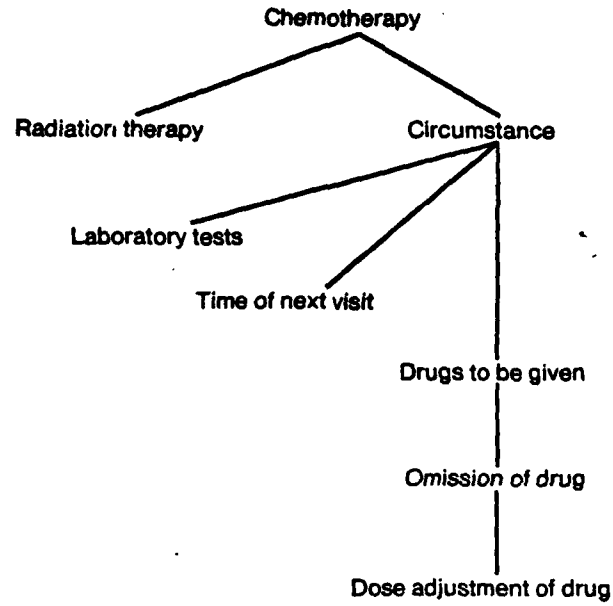


Figure 3: ONCOCIN's hierarchy of therapy planning knowledge used in the recommendation evaluation process.

a difference occurs for only *some* sub-components, they are noted as significant differences for which explanations need to be generated, but the offspring of the remaining branches are explored. The process is continued, investigating further sub-components, until each path from the top-most component either ends in a significant difference or has been explored to its fullest extent.

Consider a case of MOPP chemotherapy for Hodgkin's disease in which the doctor decides to delay therapy but ONCOCIN concludes that therapy should be given (a significant difference between the *Circumstance* parameters of the two recommendations). The analysis procedure begins by comparing the chemotherapies being used in each therapy recommendation and finds no significant difference (see Fig. 4). Since none is found, the offspring of *Chemotherapy* in the hierarchy are investigated. No significant difference is found between the *Radiation Therapy* components of the two recommendations, but the *Radiation Therapy* node has no offspring. Thus the analysis continues with *Circumstance*. Here the evaluation procedure finds a difference. Consequently, there is no need to perform any comparisons on the offspring of *Circumstance*. Since there are no other offspring of the *Chemotherapy* node of the hierarchy, the analysis process stops and the difference in *Circumstance* will be explained.

Now suppose for the same patient that ONCOCIN decides that the drug prednisone should be omitted, but the physician proposes to give treatment without omitting that drug (see Fig. 5). In this case the analysis begins as it did in the previous example, but since no difference in *Circumstance* is found, the analysis continues to all the offspring of *Circumstance* and all of their offspring without finding a significant difference. However, when the analysis process considers *Omission of a drug* for the drug prednisone, a significant difference is detected. Thus, no comparison will be made for the offspring of that node, namely the dose adjustment of prednisone. Instead the conclusion that prednisone be omitted will be explained. This corresponds to our intuitions about what should be explained in this case. ONCOCIN's conclusion that the drug should not be given is the relevant fact, not what led ONCOCIN to recommend any particular dose for the drug.

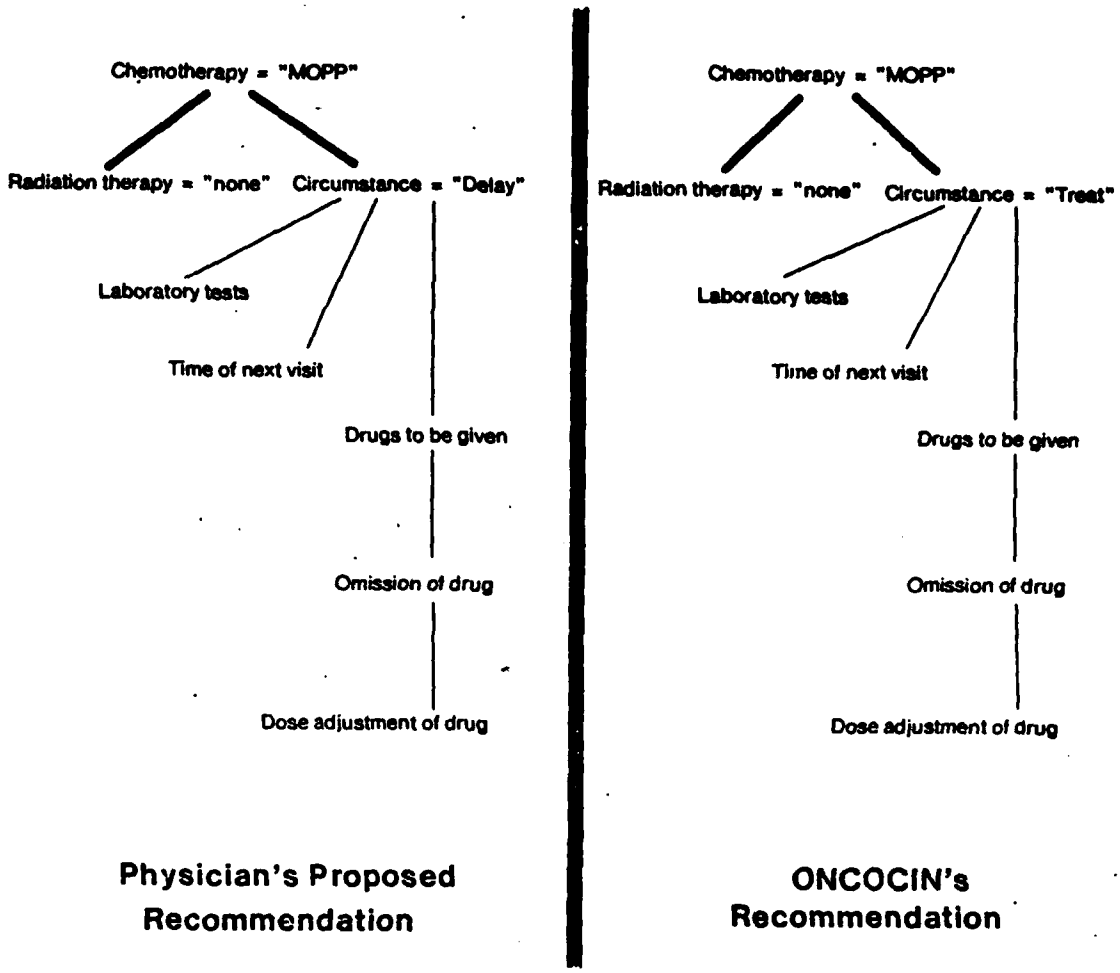


Figure 4: A schematic representation of the hierarchy overlay at termination after it has found significant differences between *Circumstance* elements of the two recommendations. The bold lines indicate those portions of the hierarchy that were searched before a significant difference was noted (see text).

Although we do not consider proposed recommendations with multiple differences in our examples above, it is important to note that the plan analysis process will find multiple differences if they are independent of one another. In contrast, since the analysis of any given branch in the hierarchy stops as soon as a significant difference is found, specific differences that occur as a result of a more general difference will not be considered separately. In particular, as the hierarchy is processed, each evaluation procedure can make the assumption that there is agreement between all components of the recommendations which occur above it in the hierarchy. For example, it makes little sense to evaluate the dosing of a particular drug recommended by the user when it was given under the assumption that a different chemotherapy was being used. Similarly it makes little sense to critique the dosing of a particular drug given by the doctor when the reason that a disagreement occurred is that ONCOCIN thought therapy should be delayed (different *Circumstance*) and did not recommend to give any drugs at all.

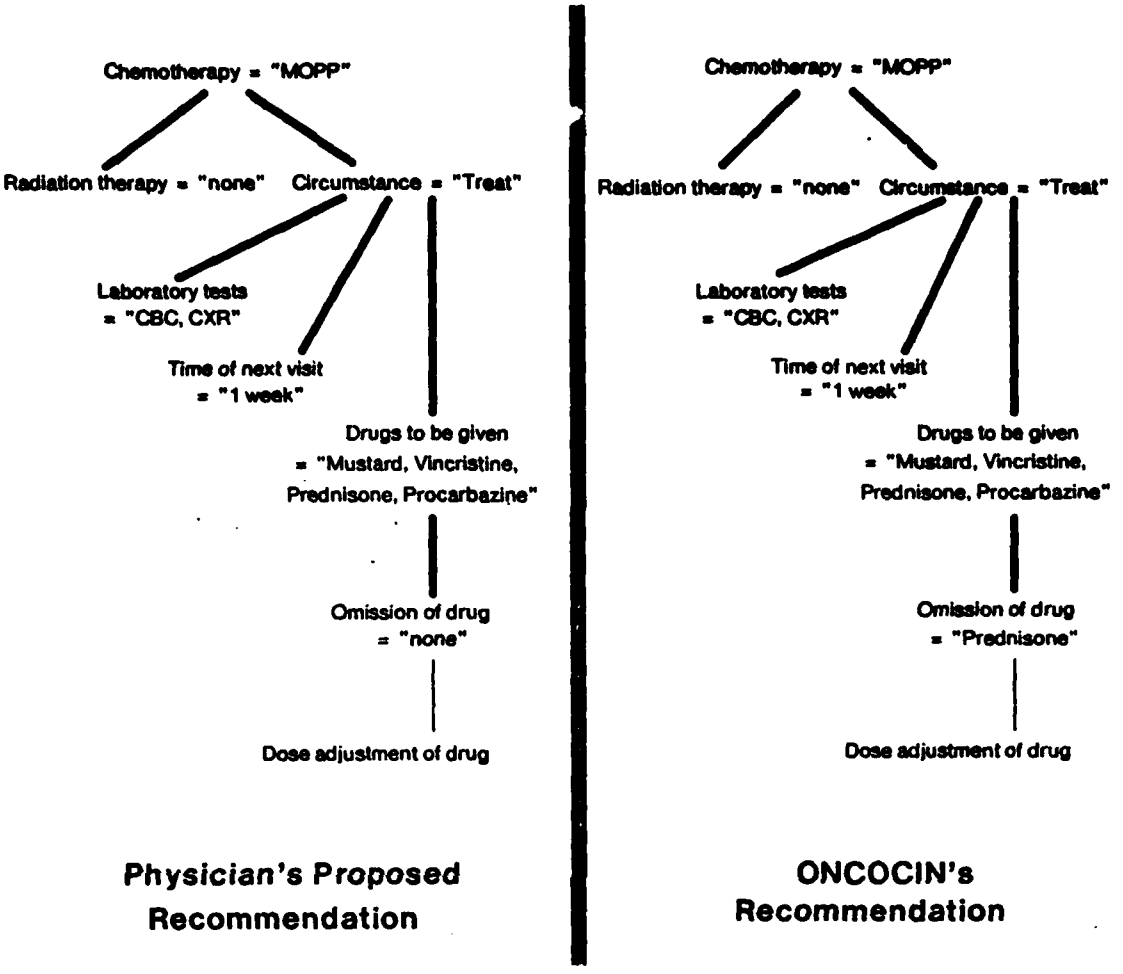


Figure 5: A schematic representation of the hierarchy overlay at termination in a case in which a significant difference between the two recommendations was not noted until the issue of drug omission was considered. Once again the bold lines indicate those portions of the hierarchy that were searched before the discrepancy was noted.

Critique Generation: Explanation of Differences

For a rule chaining system like ONCOCIN, an appropriate way to explain the program's conclusions is to allow the physician to explore a record or "trace" of the computer's reasoning in detail. Such a rule trace represents a series of links between findings (data) and treatments (goals). Teach (Teach, 1983) has shown that some physicians give significantly higher evaluations to explanations which make specific links between symptoms and diagnoses. It is presumed that the explanation of ONCOCIN's reasoning trace will help to inform users where their reasoning may have differed from that of the computer.[8] There are at least four modes of explanation used by consultation systems; our discussion will focus on the last:

[8] Underlying support and strategic knowledge will also heighten the quality of the explanations (Clancey, 1983).

1. *Explanation during a consultation* (Scott, 1977; Swartout, 1981). The user, when asked to provide data in such systems, can respond instead with a question of the form "Why do you want to know?" The system's answer provides some indication of how the datum will be used in the reasoning process.

2. *Post-consultation explanation in response to specific questions from the user* (Scott, 1977; Swartout, 1981). Post-consultation explanation methods often employ simple parsers to allow the user to ask about the knowledge used and actions taken in a particular consultation session. For example, the user might ask how the system made certain conclusions, how it used a piece of information, or what decision it made about some subproblem.

3. *Tutorial explanation* (Clancey, 1979). A tutorial dialogue may involve explanations of the *strategies* used to solve a problem. A model of the user and a means to evaluate the user's performance are important elements of a system which generates useful tutorial explanations.

4. *Explanation as a critique* (Miller, 1983). A critiquing system uses knowledge about the structure of the problem and its possible solutions to find important differences between the user's proposed solution to a problem and a computer-generated "expert" solution. These differences are used to structure the resulting explanation, called a critique.

Representation and Control Structures Necessary for Explanation

In order to generate post-consultation explanations, it is important that a system maintain a record of its reasoning and that the system have methods for explaining any part of that record. This eliminates the need to re-run the consultation for each explanation (Scott, 1977).

ONCOCIN stores a *justification* each time the value of a parameter is concluded. The justification may be the rule which caused the conclusion, or some other indication of the data structure from which the information was obtained. In addition, each time a rule is used, the situation in which it was applied is recorded. ONCOCIN stores five major types of justifications with its conclusions, and has mechanisms for explaining each:

- RULExxx: (e.g., "RULE046") indicates the rule that was invoked to cause the parameter's value to be concluded (e.g., the attenuated dose of a drug). The context in which the conclusion was made is contained on the property list of the rule as described above.

- DEFAULT: indicates that one or more rules were tried in order to find the value of the parameter, but none succeeded, so the default value of the parameter was assumed[9], (e.g., it is concluded that a drug should be given if no rules are found which can conclude that the drug should be omitted.)

- ASKED: signifies that the parameter's value was obtained by asking the user (e.g., white blood cell or platelet count).

- ALWAYSCONCLUDE: signifies the parameter's value was obtained from the domain-specific protocol knowledge (e.g., the maximum dose of a drug which can be given to a patient).

---

[9] A default rule is always tried last if no other rule has succeeded, its condition always succeeds, and its action concludes the default value. This feature can be implemented in MYCIN and other EMYCIN systems using "self-referencing" rules (Shortliffe, 1976).

- HISTORY: signifies this value was obtained from the patient's medical record (e.g., the type of tumor for which the patient is being treated). This information is entered the first time the patient is seen and need not be asked again since it is stored in the patient's file.

When a physician needs an explanation for parameters whose values are justified by the HISTORY or ALWAYSCONCLUDE flags, special purpose (domain dependent) routines are used. For example, if the user asks about a parameter whose justification is ALWAYSCONCLUDE, one such routine allows ONCOCIN to respond:

```
ONCOCIN concluded the 100% dose is 6 mg/m2 according to protocol.
Would you like to see in more detail what the protocol specifies for
     the current chemotherapy?
```

Patient history data and the specific instructions of the protocol do not lend themselves well to further explanation.

Domain Independent Explanations

Many rule-based systems determine the values of parameters using three primary means: by rules, by default, and by asking the user. Consequently, we have maintained the domain independence of the explanations for these types of conclusions.

Whenever a rule is used to conclude a parameter, it is stored as the justification for that parameter. A translation method used by Scott (Scott, 1977), called *static translation*, can be used to generate english text from the machine-readable form of such a rule. The statically translated form of one of ONCOCIN's production rules is:

```
To determine the dose attenuation due to low WBC for Nitrogen Mustard
     in MOPP, for Cytoxan in MOPP or for Cytoxan in C-MOPP:
If:  white blood count (in thousands) is between 3.5 and 4
Then:  Conclude that the dose attenuation due to low WBC is 70%
```

The static translation includes all of the situations in which the rule might apply: "for Nitrogen Mustard in MOPP, for Cytoxan in MOPP or for Cytoxan in C-MOPP." In answer to questions of the form "Why did you conclude that the dose attenuation due to low WBC is 70%?" the critiquing system uses a refinement of this technique called *dynamic translation*. The dynamically translated form of a rule contains specific values for the patient under consideration, and includes only the context in which the rule was actually used. A dynamically translated form of the same rule is:

```
To determine the dose attenuation due to low WBC for Nitrogen Mustard
     in MOPP:
Since white blood count (in thousands) (3.6) is between 3.5 and 4
It is possible to conclude that the dose attenuation due to low
     WBC is 70%
```

In contrast, the dynamic translation includes only the situation in which the rule applied to the patient under consideration, "for Nitrogen Mustard in MOPP," and it includes the actual value of the white blood count which caused the rule to succeed, "(3.6)". Since the rule has succeeded, the condition of the rule is known to be true and the action has been taken.

Default conclusions can be explained using a similar rule-related method, except that the translation of no single rule can give a complete explanation. ONCOCIN makes a default conclusion when no rules succeed to conclude a given parameter. The fact that no rules succeed indicates that at least one of the conditions failed in each one that was tried. To explain a default conclusion, ONCOCIN therefore looks through all the rules

which *could* have concluded a value for the parameter in question, and finds the part of the condition which caused each such rule to fail. These failed conjuncts can be collected, negated, and dynamically translated to form a concise explanation of why the default value was used:

```
ONCOCIN did not conclude that the patient has just had significant
radiation because the site of radiation is not:
     1) Pelvic,
     2) Mantle,
     3) Para-aortic or Inverted-Y, or
     4) Total-nodal, Sub-total-nodal or Whole-body
```

Parameters whose values which were obtained from the user are explained simply by saying so:

```
ONCOCIN concluded platelet count (in thousands) is 200 because
     that is the value that was entered.  Would you like to change it?
```

Explanation Strategies

The analysis of the physician's proposed therapy plan allows for focusing the explanation dialogue around certain conclusions. An explanation is needed only for those parameters whose values are significantly different from the corresponding ones in the user's proposed therapy plan. But it would be unacceptable to print a detailed explanation of the entire chain of reasoning for a given conclusion because it is unlikely that the user wants *all* of the information contained in a complex chain with many branches. Without a detailed model of the user and a dynamically updated store of contextual information, it is difficult to find heuristics to determine what branch of reasoning should be explained first, and in what manner. ONCOCIN avoids these difficult issues by allowing users to structure explanations to their own needs using an agenda-based system.

Explanations are printed using natural language translations generated directly from the machine-readable form of the rules[10]. As the translation process occurs, each parameter in the rule being translated is added to a list of parameters relevant to the current line of explanation called the *agenda*. After each explanation of a parameter the user can choose a parameter in the agenda to be explained next. As each new explanation is generated, the parameters in it are added to the agenda and thus become available for further explanation. The user can select another parameter in the agenda to see further explanations. In this way, the reasoning tree can be traversed in an arbitrary order according to the needs of the user (who will generally have no knowledge of ONCOCIN's control structure, parameter names, or specific rules).

ONCOCIN uses three generic types of explanations to explain to the user how it concluded the values of parameters in the agenda. Each type is designed to answer one of the following questions:

1. Why did ONCOCIN conclude what it did?

2. Why did ONCOCIN not conclude something *other* than the value that it did.

3. Why did ONCOCIN *not* conclude some specific value for a parameter?[11]

---

[10] See (Scott, 1977) for a complete description of the translation mechanism used.

[11] This is treated as a special case of 2.

An Example

The following is an exerpt from a critique in which the difference between the physician's and ONCOCIN's dose of Nitrogen Mustard was determined to be clinically significant.

```
ONCOCIN recommended:
Give Nitrogen Mustard, 8.0 mg.  IV.
     [4.2 mg./m.sq. = attenuated to 70% due to low WBC]

Whereas you recommended:
Give Nitrogen Mustard, 4.0 mg.  IV.
     [4.2 mg./m.sq. = attenuated to 35%]
```

[To begin, a simple summary of the differences is printed]

```
Would you like me to explain why ONCOCIN concluded what it did? Yes[12]

               Agenda = (ATTENDOSE)
```

[For demonstration purposes, the agenda will be printed at each iteration. This is not normally shown to the user.]

```
ONCOCIN concluded that the current attenuated dose is attenuated to
     70% due to low WBC because for Nitrogen Mustard in MOPP for
     patients with all lymphomas:
Since the patient has not received chemotherapy and the blood
     counts do warrant dose attenuation, it is possible to
     conclude that the current attenuated dose is 100% dose
     attenuated by the minimum of the dose attenuation due to
     low WBC and the dose attenuation due to low platelets.

               Agenda = (ATTENDOSE NORMALCOUNTS WBCATTEN PLTATTEN CHEMODOSE)
```

[All parameters that occurred in the rule that was just printed are added to the agenda]

```
Which of ONCOCIN's conclusions would you like explained further?
0) none
1) the current attenuated dose is attenuated
     to 70% due to low WBC
2) the blood counts do warrant dose attenuation
3) the dose attenuation due to low WBC is 70%
4) the dose attenuation due to low platelets is 100%
5) the 100% dose is 6

Choose a number> 2

ONCOCIN concluded the blood counts do warrant dose attenuation because
     the dose attenuation due to low WBC is not 100%
```

[Because the conclusion that the blood counts warrant dose attenuation was made by default, its justification is obtained by translating the negation of any conjuncts which were false. These conjuncts caused rules which might have concluded otherwise (i.e., that the blood counts do not warrant attenuation) to fail. In this case there is only one such conjunct: the fact that the dose attenuation due to low WBC is not 100%.]

```
               Agenda = (ATTENDOSE NORMALCOUNTS WBCATTEN PLTATTEN CHEMODOSE)
```

[No new parameters were introduced in the previous explanation,

[12] Those portions of the critique dialogue typed by the user are shown in underlined bold face. Commentary is shown in italics.

so the agenda remains the same.]

```
Which of ONCOCIN's conclusions would you like explained further?
0) none
1) the current attenuated dose is attenuated
     to 70% due to low WBC
2) the blood counts do warrant dose attenuation
3) the dose attenuation due to low WBC is 70%
4) the dose attenuation due to low platelets is 100%
5) the 100% dose is 6

Choose a number> 3

ONCOCIN concluded that the dose attenuation due to low WBC is 70%
     because for Nitrogen Mustard in MOPP:
Since white blood count (in thousands) (3.6) is between 3.5 and 4
It is possible to conclude that the dose attenuation due to low WBC
     is 70%
```

[Because the rule is translated dynamically, the actual value of the white blood cell count appears, and only the context which caused the rule to be tried is printed.]

```
Agenda = (ATTENDOSE NORMALCOUNTS WBCATTEN PLTATTEN CHEMODOSE WBC)
```

[The parameter "WBC" occurred in the last explanation, so it is added to the agenda.]

```
Which of ONCOCIN's conclusions would you like explained further?
0) none
1) the current attenuated dose is attenuated
     to 70% due to low WBC
2) the blood counts do warrant dose attenuation
3) the dose attenuation due to low WBC is 70%
4) the dose attenuation due to low platelets is 100%
5) the 100% dose is 6
6) white blood count (in thousands) is 3.6

Choose a number> 6

ONCOCIN concluded white blood count (in thousands) is 3.6 because that
     is the value that was entered.
Would you like to change it? No
```

[If users ask about a parameter that they entered, they are given an opportunity to change the value.]

```
Agenda = (ATTENDOSE NORMALCOUNTS WBCATTEN PLTATTEN CHEMODOSE WBC)

Which of ONCOCIN's conclusions would you like explained further?
0) none
1) the current attenuated dose is attenuated
     to 70% due to low WBC
2) the blood counts do warrant dose attenuation
3) the dose attenuation due to low WBC is 70%
4) the dose attenuation due to low platelets is 100%
5) the 100% dose is 6
6) white blood count (in thousands) is 3.6

Choose a number> 0

Would you like to change your prescribed dose of Nitrogen Mustard? ...
```

Future Work

Before the critiquing adaptation shown here becomes a regular part of ONCOCIN, it is important that both its performance and acceptability be thoroughly tested. Just as it must be demonstrated that a consultation system gives excellent advice before it is brought to a clinical setting, a critiquing system must demonstrate the ability to detect deviations from the protocol whenever experts consider them clinically significant.

Similarly, since adapting ONCOCIN to critique therapy plans involves significant changes in the computer-physician interface, it is important that these changes be tested by physicians. Physicians' feedback will be particularly important in developing a program that will conveniently accept their proposed therapy plans or their proposed changes to ONCOCIN's therapy plans.

Other ongoing work is aimed at generating explanations that will depart from simple rule translations and be more like normal discourse. Enhancing the quality of explanations is an acceptability problem whose solutions are likely to be found in AI techniques. We envision more general explanations, in which underlying support knowledge and a more detailed model of the user are used to generate statements that summarize complex chains of rules. These explanations will emphasize parts of the explanation which are most likely to be of importance to the user, while condensing or omitting areas which might be highly complex or beyond the user's level of expertise using techniques similar to those suggested by Wallis (Wallis, 1982.)

Recent innovations in AI hardware play a significant role in shaping our future research. Professional workstations will soon be available at prices similar to those currently paid by physicians who buy office computer systems. These workstations are small enough to fit conveniently into an office, and they boast high resolution graphics capabilities along with the power and speed needed to run complex AI programs. We are currently transporting the *Reasoner* to one such professional workstation and we have developed an experimental workstation version of the prototype described above. The workstation critiquing module highlights on the screen those phrases in the explanation text which correspond to parameters on the explanation agenda. These highlighted phrases can subsequently be selected for further explanation by pointing to the phrase using a movable pointer controlled by a hand-held device called a *mouse*. Thus the agenda need not be reprinted before each user choice, and more information can be shown on the screen at one time. This experiment, as well as others conducted using the workstation, indicate that graphics capabilities can be an effective means to circumvent natural language issues. It is computationally less expensive to allow users to select phrases on the screen using a mouse pointer than to interpret users' free text input. Furthermore, a dexterous user of a mouse (or similar device such as a touch screen or light pen) can participate in much more rapid interaction than is possible using a conventional computer keyboard. A user is likely to be especially intolerant of a system which requires him to wait for a response, or which requires a significant amount of typing.

Conclusion

ONCOCIN has evolved in response to comments and suggestions from physicians. User complaints about the need to override system advice have led to work which has provided us with a number of important lessons about the computing tools necessary to critique therapy plans. Of particular importance are the central roles of both plan analysis and explanation techniques. Plan analysis can indicate where significant disagreements have occurred, and thus provide information about what areas are likely to be of greatest interest to the user. This kind of information is difficult to obtain in other ways. Simple plan analysis is performed in our system using a procedure which compares component parts of two recommendations using

a group of domain-specific evaluation procedures. The most general set of differences which completely describes the significant disagreements between two recommendations is found. The analysis requires evaluation criteria for each part of a therapy plan, as well as knowledge about the hierarchical relationships of entities in a typical plan.

Another important lesson learned from our work with the critiquing model is that optimal critiquing requires sophisticated explanation techniques. Generating useful explanations consists of two separate problems: the development of the *capability to explain* a chain of reasoning in a variety of ways, and the development of techniques to give the program the *knowledge about the situation* to help it determine what explanation techniques to apply and on what knowledge structures to apply them. Considerable advances remain to be made in both of these areas. For instance, the applicability of the hierarchical plan analysis technique has not been tested in other problem domains. It should be easily adaptable to domains in which the hierarchical relationships on which the analysis depends occur frequently (as they do in clinical medicine). However, to be adapted for the wide variety of problems for which expert systems are used, the critiquing model may require more general methods for representing the structure of the problems and their possible solutions. Advances are also needed to develop additional levels of specificity in automated explanations. For example, we envision more general explanations in which strategic knowledge and more detailed information about the user lead to statements that summarize complex chains of rules.

Because of its advantages both for the user and as a computational device to obtain information about the user, we believe the critiquing model has considerably more utility than the conventional consultation model in building many interactive expert systems. Our work with ONCOCIN has shown that an expert system can be reconfigured in a manner which may significantly reduce the burden on the user. Instead of acting as a mechanized consultant that methodically asks for findings and renders a treatment decision which the physician must override if a disagreement exists, the expert clinical consultant becomes a silent partner in the decision making process and only makes its opinions known when a sub-optimal therapy is proposed by the physician. For an experienced physician these interruptions will be infrequent, and thereby less disruptive. When they occur, they offer a focused analysis of where the differences lie and why they may exist. We believe that this kind of critiquing interaction will contribute to increased acceptance of expert systems by individuals who prefer to reach decisions independently when possible.

Acknowledgements

Support for this work was provided by the National Library of Medicine under Grants LM-03395 and LM-00048, and by the Office of Naval Research under contract NR 049-479. Dr. Shortliffe is a Henry J. Kaiser Family Foundation Faculty Scholar in General Internal Medicine. Computing facilities were provided by the SUMEX-AIM resource under NIH Grant RR-00785. We also thank Bruce Buchanan for helpful comments on an earlier draft of this paper.

References

Burton, R. R. and Brown, J. S. (1979). An Investigation of Computer Coaching for Informal Learning Activities. *International Journal of Man-Machine Studies* 11: 5-24.

Clancey, W. J. (1977). An Antibiotic Therapy Selector Which Provides for Explanations. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, p. 858.

Clancey, W. J. (1979). Tutoring Rules for Guiding a Case Method Dialogue. *International Journal of Man-Machine Studies* 11: 25-49.

Clancey, W. J. and Letsinger, R. (1981). NEOMYCIN: Reconfiguring a Rule-Based Expert System for Application to Teaching. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, Vancouver, British Columbia. pp. 829-836.

Clancey, W. J. (1983). The Epistemology of a Rule-Based Expert System: A Framework for Explanation. *Artificial Intelligence*. (In press).

Duda, R. O., and Shortliffe, E. H. (1983). Expert Systems Research. *Science* 220: 261-268.

Gerring, P. F., Shortliffe, E. H., and van Melle, W. (1982). The Interviewer/Reasoner Model: An Approach to Improving System Responsiveness in Interactive AI Systems. *AI Magazine* 3(4): 24-27.

Miller, P. L. (1983). Critiquing Anesthetic Management: the "ATTENDING" Computer System. *Anesthesiology* 53: 362-369.

Patil, R. S., Szolovitz, P., Schwartz, W. B. (1981). Causal Understanding of Patient Illness in Medical Diagnosis. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, Vancouver, British Columbia. pp. 893-899.

Scott, A. C., Clancey, W. J., Davis R., and Shortliffe, E. H. (1977). Explanation Capabilities of Production-Based Consultation Systems. *American Journal of Computational Linguistics, Microfiche 62*. Also to appear in *Readings in Medical Artificial Intelligence: The First Decade* (W. J. Clancey and E. H. Shortliffe, eds.), Reading, MA: Addison Wesley, 1983.

Shortliffe, E. H. (1976). *Computer-Based Medical Consultations: MYCIN*. New York: Elsevier/North Holland.

Shortliffe, E. H., Scott, A. C., Bischoff, M., Campbell A. B., van Melle, W., Jacobs, C. (1981). ONCOCIN: An Expert System for Oncology Protocol Management. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, Vancouver, British Columbia. pp. 876-881.

Shortliffe, E. H. (1982). The Computer and Medical Decision Making: Good Advice is not Enough. *IEEE Engineering in Medicine and Biology Magazine* 1(2): 16-18. (guest editorial).

Swartout, W. R. (1981). Explaining and Justifying Expert Consulting Programs. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, Vancouver, British Columbia. pp. 815-822.

Teach, R. L. and Shortliffe, E. H. (1981). An Analysis of Physician Attitudes Regarding Computer-Based Clinical Consultation Systems. *Computers in Biomedical Research* 14: 542-558.

Teach, R. L. (1983). *Patterns of Explanation and Reasoning in Clinical Medicine.* Doctoral dissertation, Stanford University. (in preparation).

Wallis, J. W. and Shortliffe, E. H. (1982). Explanatory Power for Medical Expert Systems: Studies in the Representation of Causal Relationships for Clinical Consultations. *Methods of Information in Medicine* 21: 127-136.

Yu, V. L., Fagan, L. M., Wraith, S. M., et al. (1979). Antimicrobial Selection By A Computer: A Blinded Evaluation By Infectious Disease Experts. *Journal of the American Medical Association* 242(12): 1279-1282.

Distribution Statement

Defense Documentation Center Cameron Station Alexandria, VA 22314	12 copies
Office of Naval Research Arlington, VA 22217	
Information Systems Program (437)	2 copies
Code 200	1 copy
Code 455	1 copy
Code 458	1 copy
Office of Naval Research Eastern/Central Regional Office Bldg. 114 Section D 666 Summer St. Boston, MA 02210	1 copy
Office of Naval Research Branch Office, Chicago 536 South Clark St. Chicago, IL 60605	1 copy
Office of Naval Research Western Regional Office 1030 East Green St. Pasadena, CA 91106	1 copy
Naval Research Laboratory Technical Information Division, Code 2627 Washington, DC 20375	6 copies
Dr. A. L. Slafkosky Scientific Advisor Commandant of the Marine Corps (RD-1) Washington, DC 20380	1 copy
Naval Ocean Systems Center Advanced Software Technology Division Code 5200 San Diego, CA 92152	1 copy
Mr. E. H. Gleissner Naval Ship Research & Development Center Computation and Mathematics Department Bethesda, MD 20084	1 copy
Capt. Grace M. Hopper (D08) Naval Data Automation Command Washington Navy Yard Bldg. 166 Washington, DC 20374	1 copy
Captain R. Martin 3311 West Avenue Newport News, VA 23607	1 copy

ONR Resident Representative — 1 copy
Durand Aeronautics Bldg., Rm. 165
Stanford University
Stanford, California 94305

Edward H. Shortliffe, M.D., Ph.D. — 1 copy
Stanford Med Ctr TC-117

Michael R. Genesereth, Ph.D. — 1 copy
Computer Science Department
Stanford University